

(19)



(11)

EP 1 309 220 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
05.08.2009 Bulletin 2009/32

(51) Int Cl.:
H04Q 11/04 (2006.01) **H04J 3/16** (2006.01)
H04Q 3/00 (2006.01) **G06F 9/00** (2006.01)

(21) Application number: **01410142.2**

(22) Date of filing: **31.10.2001**

(54) Data processing system and method

Vorrichtung und Verfahren zur Datenverarbeitung

Dispositif et méthode de traitement de données

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR**

(43) Date of publication of application:
07.05.2003 Bulletin 2003/19

(73) Proprietor: **Hewlett-Packard Company**
(a Delaware Corporation)
Palo Alto CA 94304 (US)

(72) Inventor: **Lyonnaz, Gérard**
38760 Varcès (FR)

(74) Representative: **Lloyd, Richard Graham**
Hewlett Packard CCF
IP Section - Legal Department
5, Avenue Raymond Chanas-Eybens
38053 Grenoble Cedex 09 (FR)

(56) References cited:
US-A- 5 581 600

- **COHEN M L: "DSP-BASED ARCHITECTURE FOR REAL-TIME T1 BANDWIDTH MANAGEMENT AND CONTROL" INTERNATIONAL CONFERENCE ON COMMUNICATIONS. INCLUDING SUPERCOMM TECHNICAL SESSIONS. ATLANTA, APR. 15 - 19, 1990, NEW YORK, IEEE, US, vol. 2, 15 April 1990 (1990-04-15), pages 690-697, XP000146148**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 1 309 220 B1

DescriptionField of the Invention

[0001] The present invention relates to a data processing system and method and, more particularly, to a computer aided telephony system and method.

Background of the Invention

[0002] Computer aided telephony systems are well known within the art. Such systems typically include a server which runs a voice application and a digital trunk processor to provide high quality computer aided telephony functions. The telephony functions include, for example, a digital trunk adapter and E1/T1 trunk interface card, DTMF detection or generation resources, text-to-speech resources, facsimile resources and voice recognition resources.

[0003] Typically a voice application running at the server, in response to an incoming call, requires access to a number of the above computer aided telephony resources. Normally access is provided to such call processing resources via a time division multiplexed (TDM) bus such as a SONET bus. However, it is sometimes the case that the server either does not provide call processing hardware resources such as voice processing digital signal processors and the like, or does not have access to a TDM bus such as hardware and software implementing such a SONET bus. In those circumstances, it is not possible to implement a computer aided telephony system effectively.

[0004] It is an object of the present invention at least to mitigate some of the problems of the prior art.

[0005] US 5,581,600 describes a service platform for the provision of automated speech based services over a telephone network comprising a speech processor including associated memory; line interface means for terminating a number of telephony channels of the network; a speech bus to carry speech signals between the line interface means and the speech processor; a control processor to control the routing of speech signals over the speech bus; a control bus connecting the control means with the speech processor and the line interface; a separate memory holding speech processing software; and a data link connecting the memory means with the speech processing means.

[0006] Cohen, M. L: "DSP-based architecture for real-time T1 bandwidth management and control", International Conference on Communications, including supercomm technical sessions, Atlanta, APR. 15 - 19, 1990, New York, IEEE, US, vol.2, 15 April 1990 (1990-04-15), pages 690-697, XP000146148, describes a DSP-based architecture for real-time T1 bandwidth management and control.

Summary of Invention

[0007] Accordingly, a first aspect of the present invention provides a TDM bus emulation method for processing data using first and second software data processing resources of a computer aided telephony server, including a central processing unit and a memory; the first and second data processing resources each emulating a respective hardware DSP; the method comprising the steps of:

establishing, within the memory, a time slot table identifying time slots for the first and second software data processing resources; the data structure providing access to corresponding data to be processed by at least one of the first and second software data processing resources;

executing, using the central processor and memory, one of the first or second software data processing resources in a time slot indicated by a current time slot pointer for a first predetermined time period according to the data contained within the data structure to produce first data processing results from the corresponding data;

incrementing the time slot pointer;

identifying from the time slot indicated by the current time slot pointer, the next software data processing resource of the first and second data processing resources to be executed; and

executing the next software data processing resource for a second predetermined time period to process the first data processing results to produce second data processing results.

[0008] Advantageously, since the time slot table determines the order of execution of the software DSP resources and each software DSP resource is granted access to the system resources periodically or at least for a short period of time at various intervals, the conventional telephony functions required by an application can be supported even in the absence of hardware DSPs or a physical TDM bus.

[0009] It will be appreciated that it is desirable to share access to any common system resources such as the central processor and memory etc. Therefore, a preferred embodiment provides a method further comprising the step of periodically generating an interrupt which determines the duration of at least one of the first and second predetermined time periods; and terminating a currently executing software data processing resource and commencing execution of a next identified software data processing resource.

[0010] The data to be processed by a software data processing resource will be stored within the system

memory of a server. Suitably, an embodiment provides a method in which the data structure comprises for each entry at least one of an indication of an associated software data processing resource to be executed and an indication of the corresponding data to be processed by the associated software data processing resource.

[0011] Since the telephony or other functions required by an application may include a need for access to several distinct processing requirements, preferred embodiments provide a method in which a data structure is provided which comprises an indication of the next software data processing resource to which any data processing results produced by the associated software data processing resource should be directed. In this manner, the software processing resources can share or exchange data.

[0012] Preferred embodiments provide a method in which the step of establishing the time slot table comprises receiving a request from an application for access to at least one of the first and second software data processing resources.

[0013] Embodiments of the present invention find application within computer aided telephony systems. Suitably, embodiments provide a method in which at least one of the first and second software data processing resources implements a telephony related function.

[0014] At initialisation, a computer aided telephony system may not have a clear indication of the data processing resources that are available to support an application. Accordingly, an embodiment provides a method further comprising the step of interrogating system resources of the computer to identify the presence of any software data processing resources; and assigning respective identifiers to any identified software data processing resources. Suitably, the DSP resources can be used readily by an application.

[0015] It will be appreciated that a conventional TDM bus provides access to any resident DSP resources on a TDMA basis. Therefore, embodiments provide a data processing method comprising the step of stepping through each entry in the data structure and executing an identified software data processing resource using corresponding data to produce first data; which first data forms the corresponding data for a subsequently identified software data processing resource identified within the data structure. By stepping through each entry in the data structure, each of the software data processing resources is granted access to the system resources of the computer system to allow each to perform their respective functions.

[0016] An aspect of the present invention provides a data processing method comprising the steps of executing a first software data processing resource, identified in an entry of a data structure, using corresponding data to produce first data; which first data forms the corresponding data for a second software data processing resource, subsequently identified within a further entry of the data structure; and executing the second software

data processing resource to process the first data to produce second data.

[0017] A further aspect of the present invention provides a computer program element comprising computer program code for implementing a TDM bus emulation system or method.

[0018] A still further aspect of the present invention provides a computer program product comprising a computer readable storage medium having stored thereon the above described computer program element.

Brief Description of the Drawings

[0019] Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

figure 1 shows a conventional computer aided telephony system in accordance with the prior art;

figure 2 shows a computer aided telephony system in accordance with an embodiment of the present invention; and

figure 3 shows a number of data structures used in an embodiment of the present invention;

Description of the Preferred Embodiments

[0020] Referring to figure 1 there is shown a computer aided telephony system 100 comprising a data terminating equipment 102 such as, for example, a telephone or a computer. The data terminating equipment 102 is connected via a network 104 to a computer aided telephony server 106 using an E1/T1 telephony card 108. The computer aided telephony server provides a service to a customer using a combination of voice menus and speech recognition or telephone keypad keystrokes under the control of a voice application 110, which runs on top of an operating system 112.

[0021] The computer aided telephony server 106 has a number of hardware and software resources which the voice application 110 may use to service an incoming call or to establish an outgoing call. The hardware resources include an A-law or μ -law compander DSP 118 and associated software driver 120, a DTMF detector DSP 122 and associated driver 124 and a facsimile card DSP 126 and associated driver 128. Text-To-Speech and Automatic Speech Recognition DSPs and associated software drivers may also be provided (not shown).

[0022] A TDM bus 130, which operates on a time slot basis is provided to support the exchange of data between the hardware resources and the voice application. Each of the hardware DSPs can receive and output data via assigned time slots.

[0023] An incoming call is received from the data terminating equipment 102 via the E1/T1 telephony card 108 as is known with the art. The telephony card forwards

an interrupt to a voice application 110 which, in response to receiving that interrupt, calls, via the operating system 112, a telephony card DSP driver 114 that controls a hardware telephony card DSP 116 for receiving the incoming data of the E1/T1 telephony card 108.

[0024] Having received raw telephony data from the network 104, the A-law or μ -law compander DSP 118 and driver 120 expand any telephony data received from the E1/T telephony card 108 and telephony card DSP 116 and associated driver 114. The A-law or μ -law DSP 118 and driver 120 output any data processing results for further processing by the voice application or another DSP via the time division multiplexed bus 130. Once the data has been decompressed it is in a form that is suitable for further processing.

[0025] The voice application 110 can then make use of the other resources to implement desired telephony functions as is well known within the art. The DTMF detector DSP 122 and driver 124 can be used by the voice application to receive inputs from a user (not shown) of the DTE 102. If the voice application 110 needs to provide a form of fax-on-demand function, the facsimile card DSP 126 and driver 128 will be used.

[0026] Referring to figure 2 there is shown a computer aided telephony system 200 in accordance with an embodiment of the present invention. Again, there is provided a data terminating equipment 102 connected via the network 104 to a computer aided telephony server 202. The computer aided telephony server 202 comprises an E1/T1 telephony trunk card 204, a voice application 206 that runs on top of an operating system 208.

[0027] However, the computer aided telephony server 202 does not have, unlike known systems, a hardware TDM bus. Instead a TDM bus is emulated to provide a software TDM bus 210. The software TDM bus and its operation will be described in greater detail hereafter.

[0028] A number of digital signal processors which include, for example, a telephony card DSP 212, an A-law or μ -law compander DSP 214, a DTMF detector DSP 216 and a facsimile card DSP 218, should preferably be accessible using the software TDM bus 210. It will be appreciated that other DSPs can additionally be provided which perform speech recognition or synthesis functions. Each of the DSPs may be either a hardware resource or a software data processing resource that emulates a hardware DSP. In a preferred embodiment the software TDM bus 210 provides access to software resources that emulate hardware resources.

[0029] Embodiments of the present invention find particular application in implementing computer aided telephony systems using servers that do not have particular types of hardware. The particular types of hardware that are missing may include, in particular, a hardware TDM bus such as a SONET bus, or a hardware DSP resource.

[0030] A TDM bus emulator 224 is provided to emulate the operation of a conventional TDM bus, that is, the TDM bus emulator 224 implements the software TDM bus 210. The TDM bus emulator 224 supports access to the data

processing DSPs 212 to 218 by the voice application 206.

[0031] The interface between the voice application 206 and the OS 208 in the present example is substantially the same as that between the voice application and the operating system of known servers. It can be appreciated that maintaining this degree of interoperability ensures compatibility with current voice applications. In essence, the voice application will be presented with a consistent application interface to obtain support for telephony functions regardless of whether those telephony functions interact using a SONET TDM bus or a software TDM bus and software DSPs.

[0032] The TDM bus emulator 224 receives requests for access to the data processing resources and ensures that those requests are serviced so that the voice application 206 can perform its intended function. The TDM bus emulator 224 constructs sophisticated data structures to ensure that each of the data processing DSPs 212 to 218 can interact and can be accessed by the voice application 206 to perform their intended functions. The data structures are used to control the order of processing of the software DSP resources.

[0033] Referring to figure 3 there is shown an embodiment of the sophisticated data structures 300 used by the TDM bus emulator 224 in supporting the voice application 206. The data structures 300 include a DSP and function entry point table 302 which stores an identifier for each DSP, in a first field 304, and a function entry point for each of the DSPs, in a second field 306. A function entry point identifies or defines a format of the function that is used to call one of the data processing DSPs 212 to 218.

[0034] A time slot table 308 stored within the memory 220 is used by the TDM bus emulator 224 to allocate time slots to each of the DSPs identified in the first table 302. The TDM bus emulator 224 provides, in accordance with the order established in the second table 308, access to a central processor 223 or hardware resources when implementing a required telephony processing function. The central processor and other hardware resources of the server 202 are used to execute the code corresponding to the software DSPs, the application, operating system and TDM bus emulator.

[0035] The time slot table 308 comprises three columns; namely a time slot number column 310, an algorithm or DSP column 312 and a request descriptor column 314.

[0036] The time slot number column 310 contains a list of the software TDM bus time slots. The time slots that are currently in use have associated data contained within the remaining columns. In the present example, time slots 0 to 4095 are notionally available for use by the data processing resources 212 to 218.

[0037] The algorithm or DSP number column 312 comprises a DSP identifier, that is, a DSP number taken from the DSP number column 304 of the first table 302. That DSP number identifies the DSP that will be operable within the corresponding time slot. For example, it can be

seen that the second DSP, having an identifier of 1 and a function entry point FE1(...), is allowed to operate within the time slot having the number 0.

[0038] The request descriptor column 314 contains an indication of the current time slot, a start address of a data buffer within the memory 220 at which the current DSP can access data to be processed and an indication of the length of the data buffer. In the illustrated example a current time slot pointer is pointing to time slot number 0. DSP number 1, having function entry point FE1(...), is operable within time slot 0. DSP 1 will process the data that is stored within the memory at addresses XXXXh to (XXXX+11)h. The results of the data processing performed by DSP 1 are stored in a buffer beginning at address YYYYh. The DSP that is operable within the successive time slot is DSP number 2. When the current time slot pointer is incremented, in response to the TDM bus emulator having received an interrupt, DSP 2 is arranged to process the output data placed in the buffer at addresses YYYYh to (YYYY+12)h by the previous DSP, that is, DSP 1.

[0039] It can be appreciated from the above that a chain of DSPs is established and that data processed by the DSPs in that chain is passed along the chain. In essence the processing is performed in a sequence that is controlled by the position of a DSP's identifier within the time slot table.

[0040] The TDM bus emulator 224 increments the current time slot pointer every time an interrupt is received. In this manner, the TDM bus emulator steps through each of the occupied time slots and allocates processing time or resources to each of the software DSPs in turn. In a preferred embodiment, the interrupt is generated every 8 msec by the E1/T1 telephony card.

[0041] It will be appreciated by those skilled in the art that the conventional internal bus of the computer aided telephony server 202 will be used, in the event of there being a hardware DSP resource, to direct data to that hardware resource. Furthermore, it will be appreciated that, in the event of a software DSP being implemented or a software TDM bus being implemented, the conventional computing resources, such as the usual address and data buses, the central processor and on-board memory, will be made available in a multi-tasking or multi-threaded manner to allow the software DSP to execute and to provide additional telephony functionality without having to incur the overhead or inconvenience of installing further DSP hardware resources. An instance of a software DSP will be instantiated and the associated data processed using the thread or multi-tasking capability of the server.

[0042] Each DSP 212 to 218 contains a data structure which identifies the time slots on which data can be received from another DSP and on which data can be output to another DSP or returned to the application 206. The data structure also preferably contains a file descriptor column to contain a file descriptor that is used to return processing results to the application. It can be seen from

figure 3 that DSP 1 328 has within its data structure 330 both an In time slot, TS_{IN}, and an out time slot, TS_{OUT}. The variable TS_{IN} and TS_{OUT} represent or contain a time slot number of the 0 to 4095 time slots for use by the DSPs. The data structure also contains a NULL entry in the file descriptor column 332, which indicates that there is no file descriptor that can be used to return the processing results to the application. Similarly, the data structure 334 corresponding to figure DSP 2 contain the values 1 and 6 in the time slot IN and OUT columns, indicating that DSP 2 will receive data on time slot 1 and output data processing results on time slot 6. The file descriptor entry for DSP 2 contains a file descriptor, FD₂, which is used to return the data produced by DSP 2 to the application 206 during time slot 6.

[0043] At initialisation the TDM bus emulator 224 polls the hardware resources, if any, to determine their presence and to determine the processing capabilities of the server. Any located software data processing resources are loaded. The TDM bus emulator 224 assigns each of the identified processing resources a corresponding DSP number. The function entry point for each DSP is stored within the first table 302 adjacent to the assigned DSP number. The TDM bus emulator 224 can determine in advance the appropriate order of execution of each of the DSP resources and assign time slots accordingly. Alternatively, the time slots can be assigned dynamically in response to DSP processing resource calls issued by the voice application 206. In effect, for each new DSP call, the TDM bus emulator assigns appropriate IN and OUT time slots within the data structure for use by the DSP. When the time slot pointer is suitably incremented, the newly assigned time slots will be made available for use by the newly called DSP.

[0044] Embodiments that use a combination of hardware DSPs and software DSPs can be accommodated within the present invention seamlessly since, rather than the function entry point of a software DSP being stored, a call to the software driver associated with a hardware DSP is stored.

[0045] At initialisation or upon receipt of an incoming call, the current time slot pointer is set to the first time slot that contains the first DSP that is required to process the incoming call. In the example, the first DSP is a telephony card DSP 212.

[0046] The processing of data by the DSPs is performed until the processing results are returned to the application. A file descriptor within the FD column of the data structure contained within a DSP is used to indicate that the processing by the preceding sequence of DSPs or chained list of DSPs is complete and that the processing results should be returned using that file descriptor.

[0047] If more than one telephony application is launched or executed at the computer aided telephony server 202, the applications can share a common TDM bus emulator and have the processing resources divided accordingly. The DSP resources of each application will be allocated time slots within the time slot table 308 dy-

namically. Furthermore, the computer aided telephony server 202 may generally support a number of incoming and outgoing calls simultaneously. Each call can be referred to as a circuit. The processing resources required by each circuit are identified and included in the time slot map or table 308.

[0048] The present invention advantageously allows a telephony application, which would normally require a SONET bus, to be supported even in the absence of such a bus. This has the additional significant advantage that the voice application can be designed and written without regard for whether the underlying or supporting TDM technology is implemented in hardware, software or a combination of both.

[0049] All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or steps are mutually exclusive.

[0050] Each feature disclosed in this specification (including any accompanying claims, abstract and drawings), may be replaced by alternative features serving the same, equivalent or similar purpose, unless expressly stated otherwise. Thus, unless expressly stated otherwise, each feature disclosed is one example only of a generic series of equivalent or similar features.

[0051] The invention is not restricted to the details of any foregoing embodiments. The invention extends to any novel one, or any novel combination, of the features disclosed in this specification (including any accompanying claims, abstract and drawings), or to any novel one, or any novel combination, of the steps of any method or process so disclosed.

Claims

1. A TDM bus emulation method for processing data using first and second software data processing resources (212,214,216,218) of a computer aided telephony server (202), including a central processing unit (223) and a memory (220); the first and second data processing resources each emulating a respective hardware DSP; the method comprising the steps of:

establishing, within the memory, a time slot table (308) identifying time slots for the first and second software data processing resources; the data structure providing access to corresponding data to be processed by at least one of the first and second software data processing resources;

executing, using the central processor and memory, one of the first or second software data processing resources in a time slot indicated by a current time slot pointer for a first predetermined

time period according to the data contained within the data structure to produce first data processing results from the corresponding data;

incrementing the current time slot pointer; identifying, from the time slot indicated by the current time slot pointer, the next software data processing resource of the first and second data processing resources to be executed; and executing the next software data processing resource for a second predetermined time period to process the first data processing results to produce second data processing results.

2. A method as claimed in claim 1, further comprising the steps of periodically generating an interrupt which determines the duration of at least one of the first and second predetermined time periods.
3. A method as claimed in either of claims 1 and 2, in which the time slot table (308) comprises, for each entry, at least one of an indication of an associated software data processing resource to be executed and an indication of the corresponding data to be processed by the associated software data processing resource (212,214,216,218).
4. A method as claimed in claim 3, in which the time slot table (308) further comprises an indication of the next software data processing resource to which any data processing results produced by the associated software data processing resource should be directed.
5. A method as claimed in any of claims 1 to 4, in which the step of establishing the time slot table (308) comprises receiving a request from an application (206) for access to at least one of the first and second software data processing resources (212,214,216,218).
6. A method as claimed in any of claims 1 to 5, in which at least one of the first and second software data processing resources (212,214,216,218) implements a telephony function.
7. A method as claimed in any of claims 1 to 6, further comprising the steps of interrogating system resources of the computer aided telephony server (202) to identify the presence of any software data processing resources; and assigning an identifier to any identified software data processing resources.
8. A system comprising means adapted to implement all the steps of the method as claimed in any preceding claim.

9. A computer program product comprising computer program code adapted to implement all the steps of the method as claimed in any preceding claim when said program is run on an electronic device.
10. A computer program product comprising a computer readable storage medium having stored thereon a computer program element as claimed in claim 9.

Patentansprüche

1. Ein TDM-Bus-Emulationsverfahren zum Verarbeiten von Daten unter Verwendung von ersten und zweiten Software-Datenverarbeitungsressourcen (212, 214, 216, 218) eines rechnergestützten Telefonservers (202), der eine zentrale Verarbeitungseinheit (223) und einen Speicher (220) umfasst; wobei die ersten und zweiten Datenverarbeitungsressourcen jeweils einen jeweiligen Hardware-DSP emulieren; wobei das Verfahren folgende Schritte umfasst:

Einrichten einer Zeitschlitztafel (308) innerhalb des Speichers, die Zeitschlitze für die ersten und zweiten Software-Datenverarbeitungsressourcen identifiziert; wobei die Datenstruktur Zugriff auf entsprechende Daten liefert, die durch zumindest entweder die ersten oder die zweiten Software-Datenverarbeitungsressourcen zu verarbeiten sind; Ausführen entweder der ersten oder der zweiten Software-Datenverarbeitungsressourcen unter Verwendung des zentralen Prozessors und des Speichers in einem Zeitschlitz, der durch einen Aktueller-Zeitschlitz-Zeiger für eine erste vorbestimmte Zeitperiode angezeigt wird, gemäß den Daten, die in der Datenstruktur enthalten sind, um erste Datenverarbeitungsergebnisse von den entsprechenden Daten zu erzeugen; Inkrementieren des Aktueller-Zeitschlitz-Zeigers; Identifizieren der nächsten Software-Datenverarbeitungsressource der ersten und zweiten Datenverarbeitungsressourcen, die zu verarbeiten ist, von dem Zeitschlitz, der durch den Aktueller-Zeitschlitz-Zeiger angezeigt wird; und Ausführen der nächsten Software-Datenverarbeitungsressource für eine zweite vorbestimmte Zeitperiode, um die ersten Datenverarbeitungsergebnisse zu verarbeiten, um zweite Datenverarbeitungsergebnisse zu erzeugen.

2. Ein Verfahren gemäß Anspruch 1, das ferner folgenden Schritt umfasst:

periodisches Erzeugen einer Unterbrechung, die die Dauer von zumindest entweder der er-

sten oder der zweiten vorbestimmten Zeitperiode bestimmt.

3. Ein Verfahren gemäß Anspruch 1 oder 2, bei dem die Zeitschlitztafel (308) für jeden Eintrag zumindest entweder eine Anzeige einer zugeordneten Software-Datenverarbeitungsressource, die auszuführen ist, oder eine Anzeige der entsprechenden Daten enthält, die durch die zugeordnete Software-Datenverarbeitungsressource (212, 214, 216, 218) zu verarbeiten sind.

4. Ein Verfahren gemäß Anspruch 3, bei dem die Zeitschlitztafel (308) ferner eine Anzeige der nächsten Software-Datenverarbeitungsressource umfasst, an die alle Datenverarbeitungsergebnisse, die durch die zugeordnete Software-Datenverarbeitungsressource erzeugt werden, geleitet werden sollten.

5. Ein Verfahren gemäß einem der Ansprüche 1 bis 4, bei dem der Schritt des Einrichtens der Zeitschlitztafel (308) das Empfangen einer Anforderung von einer Anwendung (206) für Zugriff auf zumindest entweder die ersten oder die zweiten Software-Datenverarbeitungsressourcen (212, 214, 216, 218) umfasst.

6. Ein Verfahren gemäß einem der Ansprüche 1 bis 5, bei dem zumindest entweder die ersten oder die zweiten Software-Datenverarbeitungsressourcen (212, 214, 216, 218) eine Telefonfunktion implementieren.

7. Ein Verfahren gemäß einem der Ansprüche 1 bis 6, das ferner folgende Schritte umfasst:

Abfragen von Systemressourcen des rechnergestützten Telefonservers (202) zum Identifizieren des Vorliegens jeglicher Software-Datenverarbeitungsressourcen; und Zuweisen eines Identifizierers zu jeder identifizierten Software-Datenverarbeitungsressource.

8. Ein System, das eine Einrichtung umfasst, die angepasst ist, um alle Schritte des Verfahrens gemäß einem der vorhergehenden Ansprüche zu implementieren.

9. Ein Computerprogrammprodukt, das einen Computerprogrammcode umfasst, der angepasst ist, um alle Schritte des Verfahrens gemäß einem der vorhergehenden Ansprüche zu implementieren, wenn das Programm auf einem elektronischen Gerät läuft.

10. Ein Computerprogrammprodukt, das ein computerlesbares Speichermedium umfasst, auf dem ein Computerprogrammelement gemäß Anspruch 9 ge-

speichert ist.

Revendications

1. Méthode d'émulation de bus TDM pour traiter des données en utilisant des première et deuxième ressources de traitement de données logicielles (212, 214, 216, 218) d'un serveur de téléphonie assisté par ordinateur (202), comprenant une unité centrale (223) et une mémoire (220) ; les première et deuxième ressources de traitement de données émulant chacune un DSP matériel respectif ; la méthode comprenant les étapes consistant à :

■ établir, à l'intérieur de la mémoire, une table d'intervalles de temps (308) identifiant des intervalles de temps pour les première et deuxième ressources de traitement de données logicielles ; la structure de données fournissant l'accès aux données correspondantes à traiter par au moins l'une des première et deuxième ressources de traitement de données logicielles ;

■ exécuter, en utilisant l'unité centrale et la mémoire, l'une des première et deuxième ressources de traitement de données logicielles dans un intervalle de temps indiqué par un pointeur d'intervalle de temps actuel pour une première période de temps prédéterminée en fonction des données contenues dans la structure de données pour produire des premiers résultats de traitement de données à partir des données correspondantes ;

■ incrémenter le pointeur d'intervalle de temps actuel ;

■ identifier, à partir de l'intervalle de temps indiqué par le pointeur d'intervalle de temps actuel, la ressource de traitement de données logicielle suivante parmi les première et deuxième ressources de traitement de données à exécuter ; et

■ exécuter la ressource de traitement de données logicielle suivante pendant une deuxième période de temps prédéterminée pour traiter les premiers résultats de traitement de données pour produire des deuxième résultats de traitement de données.

2. Méthode selon la revendication 1, comprenant en outre l'étape consistant à générer périodiquement une interruption qui détermine la durée d'au moins l'une des première et deuxième périodes de temps prédéterminées.

3. Méthode selon l'une quelconque des revendications 1 et 2, dans lequel la table d'intervalles de temps (308) comprend, pour chaque entrée, au moins l'une

d'une indication d'une ressource de traitement de données logicielle associée à exécuter et d'une indication des données correspondantes à traiter par la ressource de traitement de données logicielle associée (212, 214, 216, 218).

4. Méthode selon la revendication 3, dans lequel la table d'intervalles de temps (308) comprend en outre une indication de la ressource de traitement de données logicielle suivante vers laquelle des résultats de traitement de données produits par la ressource de traitement de données logicielle associée doivent être dirigés.

5. Méthode selon l'une quelconque des revendications 1 à 4, dans lequel l'étape d'établissement de la table d'intervalles de temps (308) comprend l'étape consistant à recevoir une demande d'une application (206) pour accéder à au moins l'une des première et deuxième ressources de traitement de données logicielles (212, 214, 216, 218).

6. Méthode selon l'une quelconque des revendications 1 à 5, dans lequel au moins l'une des première et deuxième ressources de traitement de données logicielles (212, 214, 216, 218) met en oeuvre une fonction de téléphonie.

7. Méthode selon l'une quelconque des revendications 1 à 6, comprenant en outre les étapes consistant à :

■ interroger les ressources de système du serveur de téléphonie assisté par ordinateur (202) pour identifier la présence de ressources de traitement de données logicielles ; et

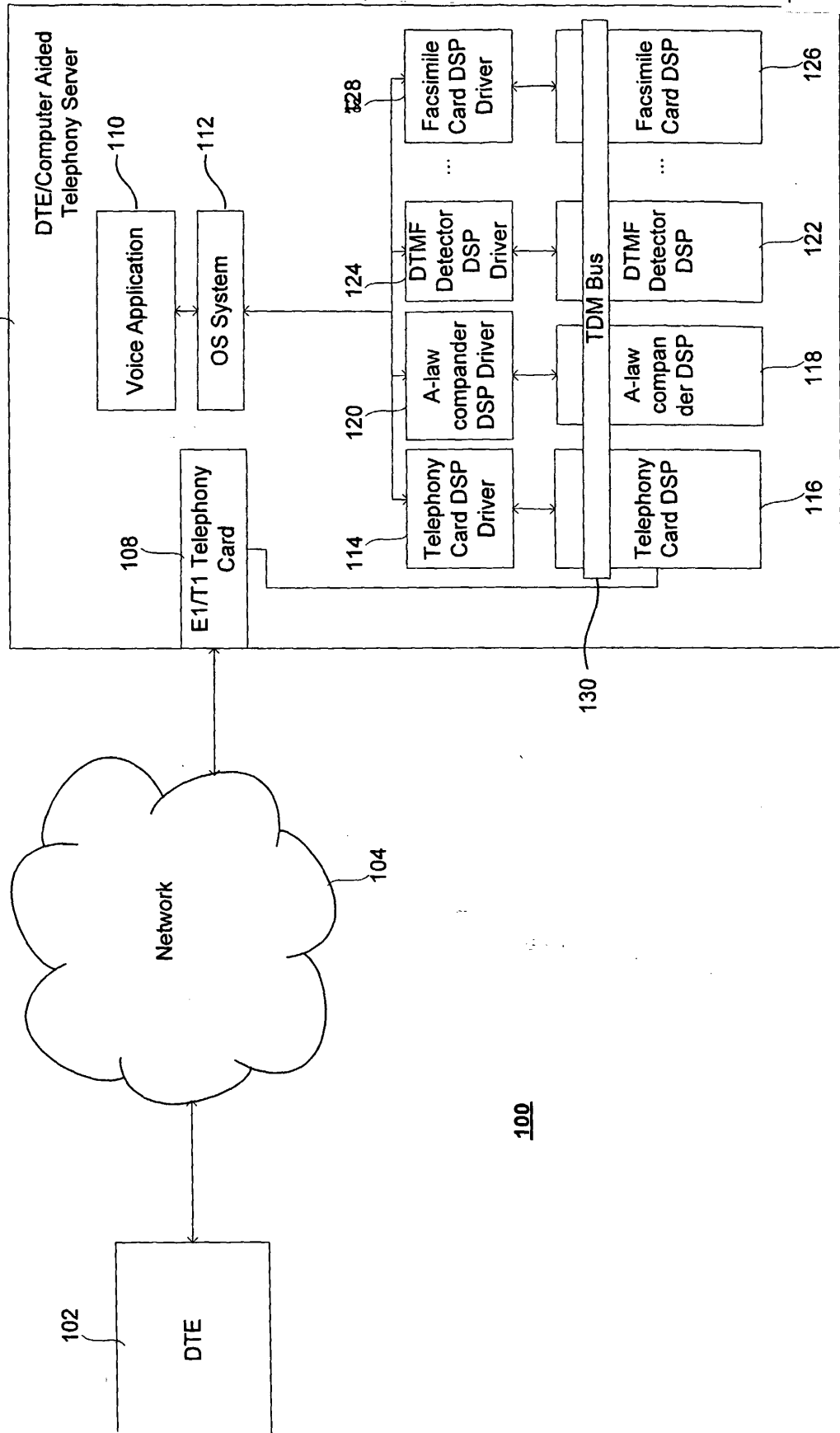
■ attribuer un identifiant à toute ressource de traitement de données logicielle identifiée.

8. Système comprenant des moyens aptes à mettre en oeuvre toutes les étapes de la méthode selon l'une quelconque des revendications précédentes.

9. Produit de programme informatique comprenant un code de programme informatique apte à mettre en oeuvre toutes les étapes de la méthode selon l'une quelconque des revendications précédentes lorsque ledit programme est exécuté sur un dispositif électronique.

10. Produit de programme informatique comprenant un support de stockage lisible par ordinateur sur lequel est stocké un élément de programme informatique selon la revendication 9.

Figure 1



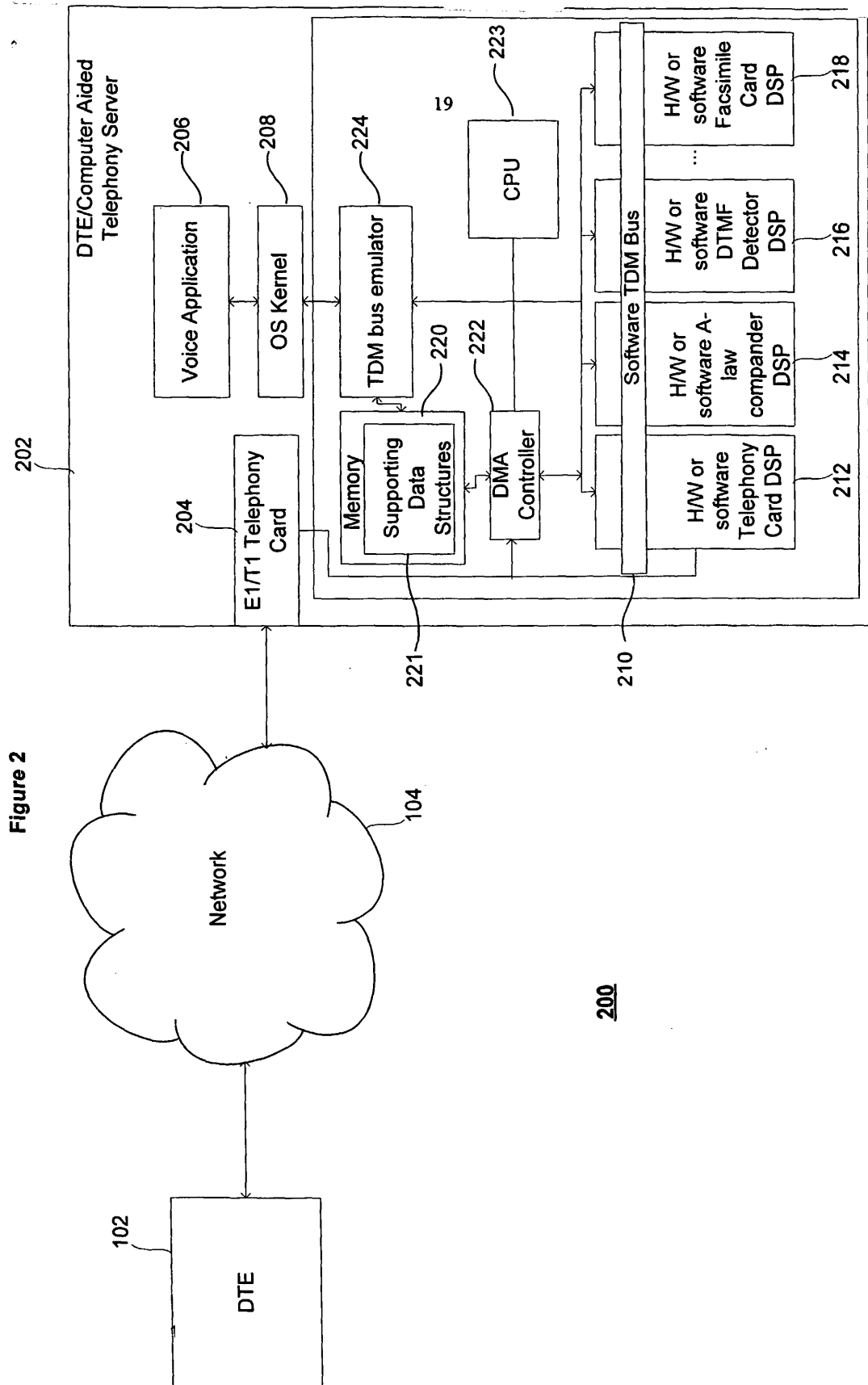
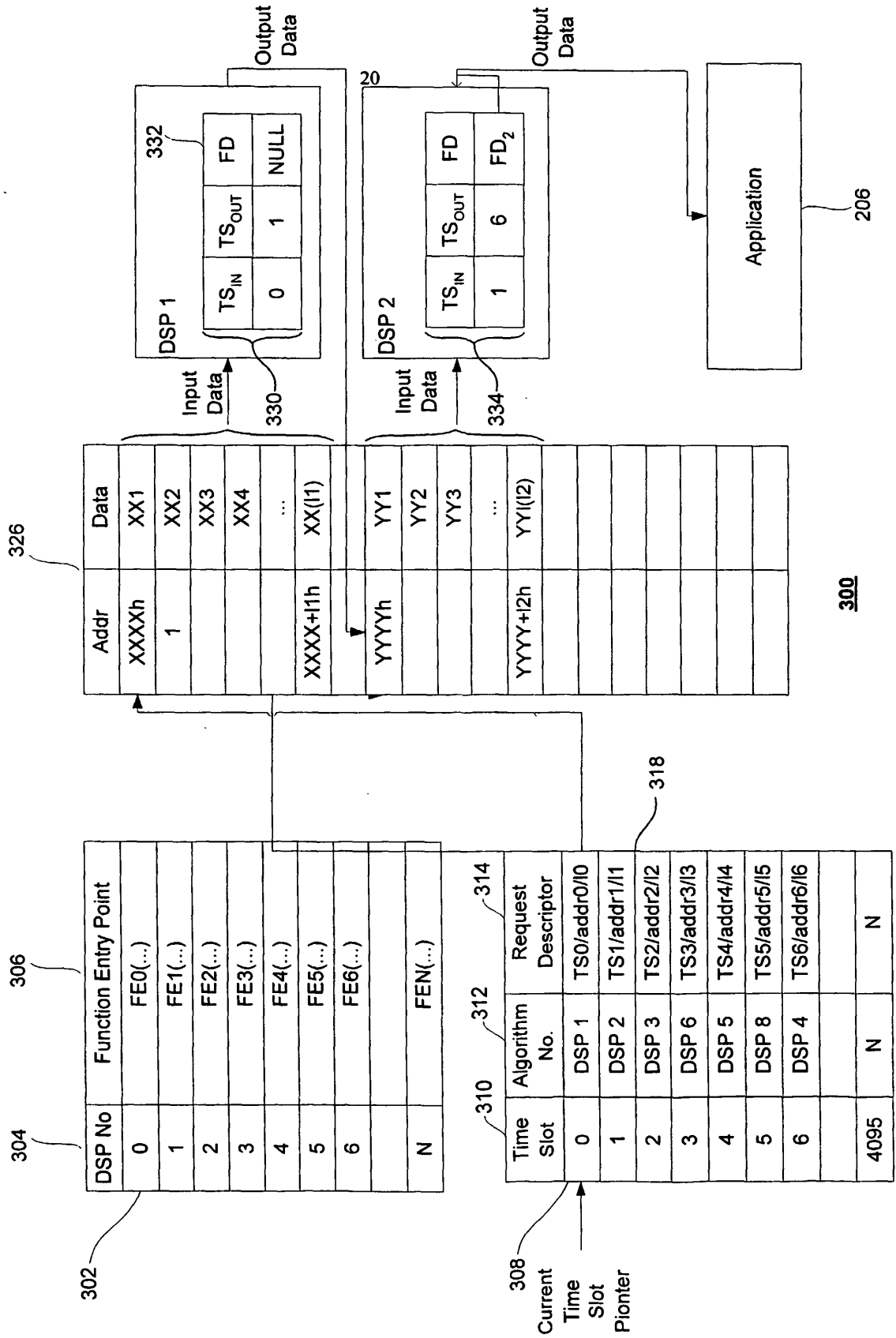


Figure 3



REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 5581600 A [0005]

Non-patent literature cited in the description

- **Cohen, M. L.** DSP-based architecture for real-time T1 bandwidth management and control. *International Conference on Communications, including super-comm technical sessions*, 15 April 1990, vol. 2, 690-697 [0006]